

Readme

Content and SDK Overview

Table of Contents

Revision History.....	3
Introduction	4
Purpose of this document.....	4
DSEA	4
Developer Portal	4
Abbreviations	4
Overview	5
SDK Service.....	5
End-user Installation	5
SDK Agent.....	5
Startup.....	5
SDK Library	6
Web SDK.....	6
WebSocket	6
Web HID	6
Compatibility.....	7
Content.....	8
Root Folder.....	8
Demo App	8
Validation	9
SDK Library	9
SDK Service.....	9
EPOS Connect.....	10
EPOS Manager.....	10
EPOS Busylight	10
Support.....	10
Features.....	11
Call States.....	11
Presence States	11
Custom Presence State	11
IM Notification	11

Presenting	11
------------------	----

Revision History

Date	Version	Change
January 09, 2022	1.0	First release
August 29, 2022	1.1	Updated Busylight states

Introduction

Purpose of this document

This document provides an overview over the different Call Control (CC) SDK offered by EPOS and when each should be used.

DSEA

DSEA Group is a subsidiary of EPOS Group. All DSEA products are developed and maintained by EPOS Group.

Developer Portal

To get the latest version of EPOS or DSEA branded software please visit the EPOS Developer Portal at this [link](#) or sign up at this [link](#).

Abbreviations

Abbreviation	Description
API	Application Programming Interface
BL	Busylight
BT	Bluetooth
CC	Call Control
HID	Human Interface Device
SDK	Software Development Kit

Overview

This section will provide an overview of the different SDK offered by EPOS.

More details and how to use and integrate the SDK can be seen in the documentation that is included in the respective SDK folder.

All SDK support full call control functionality and Busylight control.

SDK Service

The SDK Service is a process that runs in the background on either Windows or MacOS. It will expose several interfaces that a softphone can use for easy integration of call control with EPOS devices. The SDK Service must be installed on the end-user's machine.

It is recommended to use the SDK Library instead of the SDK Service if possible. The library is a newer solution that is optimized for easy integration and does not require a separate installation and background process to work.

The SDK Service is still maintained and can be used with complete confidence in quality, features and security.

End-user Installation

The installation of the SDK Service to the end-user can be done in 2 ways:

- The SDK installer can be integrated into the softphone installer and executed as a silent installation. Alternatively, the softphone can run the SDK installation the first time an EPOS device is detected.
- The SDK Service is included in EPOS Connect installation (see section below for more info on EPOS Connect). The end-user can download and install EPOS Connect directly from the EPOS webpage. This will install the SDK Service as well.

Please note that the SDK installer is shared under NDA and may not be distributed directly to end-users or customers.

SDK Agent

The installation creates 2 executables: an SDK Agent and the SDK Service. The Agent will automatically launch the SDK Service and make sure that the Service will stay healthy and active.

Only the SDK Agent should be started to ensure optimal performance.

Startup

The SDK Agent will automatically launch at system start-up. It can be a good idea for the softphone to launch the Agent when logging-in since the end-user can disable automatic launch at

system start-up. Only 1 Agent can run at any time. This means that if 1 Agent is started when an Agent is already running, it will automatically exit.

SDK Library

The SDK Library is a native cross-platform c++ library that can be used for easy integration of call control with EPOS devices. The library supports Windows, MacOS and Linux.

The library does not have any external dependencies and will work without additional effort.

Refer to SDK documentation for info on OS and hardware/architecture support.

Web SDK

The Web SDK is a JavaScript file that exposes functions for easy integration of call control with EPOS devices.

This SDK supports EPOS devices via either WebSocket or Web HID.

WebSocket

The Web SDK will attempt to establish a WebSocket connection the SDK Service which will enable call control with EPOS devices.

The SDK Service must be running on the end-user's machine which can be done by installing EPOS Connect that includes the SDK Service.

Web HID

If the Web SDK cannot create a WebSocket connection to the SDK Service, it will use Web HID to communicate with EPOS devices on supported browsers.

Web HID contains full support for call control and Busylight but cannot handle simultaneous calls on dual softphone.

Supported browsers can be seen here [link](#).

Compatibility

The below table gives a brief overview of the compatibility of the different SDK. More details can be found in the documentation that is included in the folder of each SDK.

Platform	SDK Service	SDK Library	Web SDK ³
Windows 7	✓	✓	
Windows 10	✓	✓	
Windows 11	✓	✓	
MacOS	✓	✓	
Linux ¹		✓	
VDI ²		✓	
Google Chrome			✓
Microsoft Edge			✓

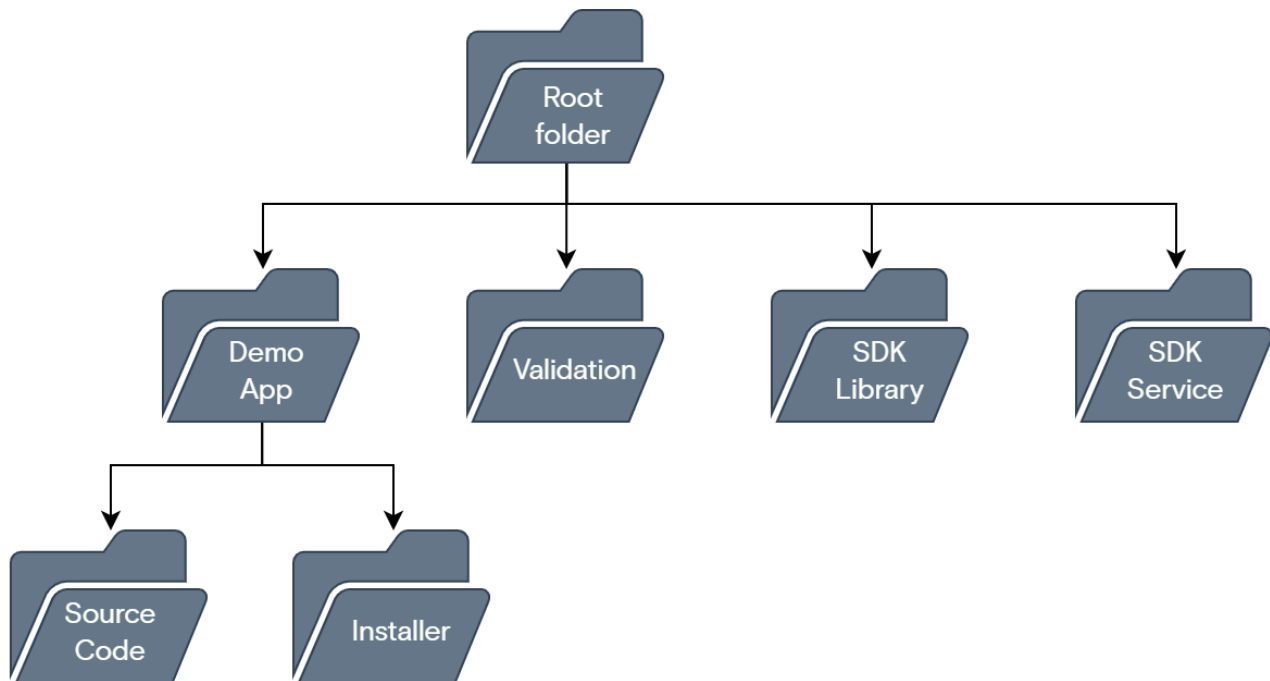
¹ Verified Linux systems are Ubuntu 18, Ubuntu 20.04, Fedora 34.

² Verified VDI systems are HP ThinPro, Dell Wyse, eLuX and IGEL.

³ Web SDK is verified on Google Chrome and Microsoft Edge with both Web HID and WebSocket.

Content

The content of the zip file downloaded from the Developer Portal will follow the structure as shown in the diagram below with small variations across the different platforms.



Root Folder

The Root folder contains this document.

Demo App

The DSEA Demo App is a desktop application on Windows (supported on Win 10 and above) and Mac. Additionally, there is a web version of the Demo App in the form of a HTML file. There is currently no Demo App on Linux.

The Demo App is provided to help testing and showcasing how to integrate the SDK into a softphone. It will allow for a fast and efficient SDK integration with both an installer and the source code provided.

Both the SDK Library and SDK Service are included in the Demo App desktop installer to enable easy testing of their API and EPOS devices.

Please note that the source code and Demo App are provided under NDA and cannot be shared externally.

Validation

The folder contains a Test Matrix that can be used to validate integration of an SDK. It contains both simple and advanced scenarios for call control and Busylight features.

It is recommended to execute the relevant test cases before deploying to marked to ensure high quality with EPOS devices.

SDK Library

This folder contains the native c++ Library on Windows, MacOS and Linux. On Web it contains the js file.

Relevant documentation such as User Guide and API description are included.

SDK Service

This folder contains the installer of the SDK Service and User Guide documentation.

EPOS Connect

EPOS Connect is a desktop application for Windows and Mac OS that provides the latest firmware and personalization options for DSEA headsets and Busylight device. It also provides Enterprise Management of the devices along with call statistics reporting to ensure flawless operation and maximum productivity. The latest version of EPOS Connect can be downloaded at this [link](#).

The EPOS Connect installer includes the SDK Service. EPOS Connect will automatically launch at system start-up and will launch the SDK Service as well.

EPOS Manager

EPOS Manager is a management tool that gives control over EPOS devices across locations, enable updates on your own terms, provide metrics and more

Settings in EPOS Connect can be managed and locked from EPOS Manager. This can be used to change the default colors of the Busylight for all end-users.

More information about EPOS Manager can be found at EPOS website at this [link](#).

EPOS Busylight

The EPOS USB Busylight increases the productivity of the office workers. For example, it makes visible when somebody is busy because they are in a call or because they are focusing on an activity which requires no interruptions. The EPOS USB Busylight also provides a ringer that notifies incoming calls, ensuring that no call is accidentally missed.

More information about the EPOS USB Busylight can be found at EPOS web site at this [link](#).

Support

The Busylight is supported in SDK Service, SDK Library and Web SDK.

The colors of all Busylight states can be modified in EPOS Connect or EPOS Manager. Without EPOS Connect the colors will use the default values shown in the table below:

Busylight State	Feature	Priority	Default Color	Behavior	Ringling
Incoming	Call State	1	Red	Flashing	Yes
Outgoing	Call State	2	Red	Flashing	No
In a Call + Presenting	Call State + Presenting flag	3	Purple	On	No
In a Call	Call State	4	Red	On	No
IM Notification	IM Notification	5	Blue	Flash for 10 seconds then returns to previous state	No
Custom Presence	Presence State	6	Custom Color	Custom Behavior	No
Available	Presence State	6	Green	On	No
Away	Presence State	6	Yellow	On	No

Busy	Presence State	6	Red	On	No
Do Not Disturb	Presence State	6	Blue	On	No

The Busylight state with the highest priority will be shown.

Features

The Busylight is supported in several ways. Call states are handled by the SDK while Presence states, IM Notification and Presenting are exposed by the SDK API and requires softphone integration.

Call States

Busylight is activated on incoming, outgoing, active and hold calls. This is managed from the SDK (Service, Library and Web) and will happen automatically when a softphone sends one of the call states mentioned before to the SDK. The softphone does not need to do anything extra to make it work.

Presence States

There are different presence states. Supported states are: Available, Away, Busy and DoNotDisturb. These can be set from a softphone via the SDK API. Refer to SDK documentation for more info.

Custom Presence State

Additional to the predefined presences states, the SDK allows the softphone to set a custom presence state. Both the color and behavior (LED flashing pattern) can be freely customized.

IM Notification

This event is set from the softphone via the SDK API. When the SDK receives this event, the Busylight will flash for 10 seconds and then return to the previous state.

Presenting

This is a flag that is set from the softphone via the SDK API. It will only be shown when the Busylight is "In a Call".